Function-on-Function Bayesian Optimization

Jingru Huang[1], Haijie Xu[1], Manrui Jiang[1], Chen Zhang[1*]

[1]Department of Industrial Engineering, Tsinghua University, Beijing 100084, China.
jingruhuang@tsinghua.edu.cn, xu-hj22@mails.tsinghua.edu.cn,
jiangmanrui@mail.tsinghua.edu.cn, zhangchen01@tsinghua.edu.cn

Abstract

Bayesian optimization (BO) has been widely used to optimize expensive and gradient-free objective functions across various domains. However, existing BO methods have not addressed the objective where both inputs and outputs are functions, which increasingly arise in complex systems as advanced sensing technologies. To fill this gap, we propose a novel function-on-function Bayesian optimization (FFBO) framework. Specifically, we first introduce a function-on-function Gaussian process (FFGP) model with a separable operator-valued kernel to capture the correlations between function-valued inputs and outputs. Compared to existing Gaussian process models, FFGP is modeled directly in the function space. Based on FFGP, we define a scalar upper confidence bound (UCB) acquisition function using a weighted operator-based scalarization strategy. Then, a scalable functional gradient ascent algorithm (FGA) is developed to efficiently identify the optimal function-valued input. We further analyze the theoretical properties of the proposed method. Extensive experiments on synthetic and real-world data demonstrate the superior performance of FFBO over existing approaches.

Code — https://github.com/aaai26-22243/Function-on-Function-Bayesian-Optimization

Introduction

Bayesian optimization (BO) is a sample-efficient framework for optimizing expensive, black-box, and gradient-free functions (Frazier 2018; Wang et al. 2023). It has been widely applied in areas such as hyperparameter tuning (Snoek, Larochelle, and Adams 2012), robotics (Wang et al. 2022), and engineering design (Shields et al. 2021). A classical BO framework consists of two main steps. The first step constructs a probabilistic surrogate model, typically a Gaussian process (GP), that approximates the underlying function. The second step involves sequentially selects query points by maximizing an acquisition function, such as the upper confidence bound (UCB) or expected improvement (EI), by trading off exploration and exploitation. Existing BO methods can be broadly divided into four types based on the output-input structure: scalar-on-vector BO (Snoek, Larochelle, and Adams 2012; Daxberger et al. 2020), scalar-on-function BO (Vien, Zimmermann, and Toussaint 2018), vector-on-vector BO (Khan, Goldberg, and Pelikan 2002; Laumanns and Ocenasek 2002), and function-on-vector BO (Huang et al. 2021).

Recently, complex systems involving function-valued data have attracted increasing attention across domains such as manufacturing (Tuo et al. 2023), marketing (Jank and Zhang 2011), and transportation (Lan et al. 2023), where it is often necessary to optimize function-valued inputs to guide the system toward the desired function-valued outputs (Chen et al. 2021). In this context, BO provides a promising approach because systems with function-valued data are usually expensive to evaluate. However, to the best of our knowledge, no existing work has studied function to function Bayesian optimization (FFBO), where both the inputs and outputs are functions defined over infinite-dimensional function spaces. A motivating example is the design of a 3D printed aortic valve constructed from tissue mimicking metamaterials. By adjusting the shape of the meta-tissue, the mechanical properties of the printed valve can be aligned with those of a real biological organ. In this setting, sine wave curves represent the shapes of the sinusoidal metamaterials and serve as function-valued inputs, while the corresponding stress strain curves represent function-valued outputs that characterize the mechanical behavior of the printed valves. However, each prototype must be printed and mechanically tested, and this process requires several hours. To efficiently identify the optimal metamaterial design under limited experimental resources and to accurately model this function on function system, an effective FFBO framework based on an FFGP surrogate model is needed. Developing such an FFBO framework is therefore of great importance, yet it introduces two major challenges.

The first challenge lies in constructing a GP surrogate model over a probabilistic space with both function-valued inputs and outputs. While GPs have been extensively studied for scalar and vector inputs and outputs, modeling function-valued inputs or outputs remains a relatively recent direction (Han and Ouyang 2021; Wang, Ng, and Haskell 2022). For GPs with function-valued inputs, prior works have introduced various distance based similarity measures to define stationary kernels, such as the L^2 norm (Mor-

*Corresponding author.

ris 2012; Nguyen and Peraire 2015), the L^p norm (Sung et al. 2024), and the spectral norm (Chen et al. 2021). For GPs with function-valued outputs, existing methods typically discretize the outputs over a fixed grid, thereby reducing the modeling problem to a multivariate GP (Han and Ouyang 2021). However, such discretization strategies face several limitations. First, in many real-world applications, data are observed at irregular time points. Interpolation is then required to align the functions onto a common grid, which may introduce additional noise, especially when the outputs exhibit complex structures or sharp local variations. Second, dense discretization leads to high-dimensional outputs, resulting in significant computational and memory burdens during inference. These challenges highlight the need for a GP model that can directly handle continuous function-valued inputs and outputs without relying on discretization.

The second challenge is to define a scalar acquisition function that transforms the functional optimization into a tractable scalar one, and then identify the optimal query over a function-valued input space. For BO with function-valued inputs, Vien, Zimmermann, and Toussaint (2018) proposes a function-valued input BO method by assuming that the function-valued inputs lie in a reproducing kernel Hilbert space (RKHS), and the acquisition function is optimized through its Fréchet derivative within the RKHS. Gultchin et al. (2023) extends function-valued input BO to causal graphs with function-valued interventions. They define interventions in an RKHS and identify them within a low-dimensional subspace by maximizing improvement over the causal graph. Astudillo and Frazier (2021) further studies BO over function-valued inputs in a functional network, where function-valued inputs are discretized on a fixed grid and treated as vectors, and multi-output BO is then applied to identify the optimal intervention. For BO with function-valued outputs, Huang et al. (2021) is the only related work. It proposes an L^2-based scalarization of the output and approximates it using functional principal component analysis (FPCA), which reduces the optimization to a finite set of principal component scores. All these discretization-based methods suffer from limited approximation accuracy and consequently lead to compromised optimization results. These limitations create a strong demand for a reasonable acquisition function and scalable optimization strategies that can directly query optimal inputs in a continuous function space.

In this work, we propose the first function-on-function Gaussian process (FFGP) model, which is defined directly over the function space. Building upon this model, we define a scalar acquisition function through a weighted scalarization strategy and introduce a functional gradient ascent (FGA) algorithm to efficiently identify optimal inputs in a function space. Our main contributions are summarized as follows:

- We propose a novel FFGP surrogate model for systems with both function-valued inputs and outputs. It is constructed by using a separable operator-valued kernel to capture the covariance structures between functions in infinite-dimensional spaces.
- Based on the FFGP surrogate, we develop the first function-to-function Bayesian optimization (FFBO) algorithm. A scalar upper confidence bound (UCB) acquisition function is defined via a weighted operator-based scalarization strategy, which transforms the functional optimization problem into a tractable scalar form. We further design an FGA algorithm that directly queries the next input function in an infinite-dimensional input space.
- We provide theoretical analysis of the proposed FFGP and FFBO frameworks. Extensive synthetic experiments and a real-world case study demonstrate the effectiveness and superiority of our method compared to state-of-the-art baselines.

The remainder of this paper is organized as follows. Section 2 formalizes the FFBO problem. Section 3 introduces the FFGP framework. Section 4 presents the FFBO algorithm based on the FFGP model. Section 5 provides theoretical guarantees. Section 6 evaluates the proposed method through synthetic and real-world experiments. Section 7 concludes the paper.

Problem Formulation

In this work, we consider the problem of optimizing an expensive-to-evaluate and gradient-free function $f : \mathcal{X}^p \to \mathcal{Y}$, where the input $\boldsymbol{x} \in \mathcal{X}^p$ is a p-dimensional vector of functions, and the output $f(\boldsymbol{x}) \in \mathcal{Y}$ is also a function. To transform this function-on-function optimization problem into a scalar one, we introduce a linear bounded operator $L_\phi \in \mathcal{L}(\mathcal{Y}, \mathbb{R})$, defined as

$$L_\phi f(\boldsymbol{x}) := \int_{\Omega_y} \phi(t) f(\boldsymbol{x})(t) dt, \tag{1}$$

where $\phi(t)$ is a user-specified or learnable weight function. Common choices include Dirac delta functions, smoothing kernels, and uniform weights (Paria, Kandasamy, and Póczos 2020; Chowdhury and Gopalan 2021). The goal is to identify the optimal input that maximizes the scalarized function-valued output, formulated as the following optimization problem:

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x} \in \mathcal{X}^p} L_\phi f(\boldsymbol{x}). \tag{2}$$

To solve this problem, we aim to develop an efficient FFBO framework. We assume an initial dataset $\mathcal{D}_0 = \{\mathbf{X}_n, \boldsymbol{Y}_n\}$, where $\mathbf{X}_n = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ and $\boldsymbol{Y}_n = (y_1, \ldots, y_n)^\top$ denote the observed input-output pairs. The observations are collected as

$$y_i = f(\boldsymbol{x}_i) + \varepsilon_i, \quad \forall i = 1, 2, \ldots, n, \tag{3}$$

where the noises $\{\varepsilon_i\}_{i=1}^n$ are assumed to be independent and identically distributed functional Gaussian random variables with $\mathrm{N}(0, \tau^2 I_{\mathcal{Y}})$.

To be general, we assume that both function-valued inputs and outputs are square-integrable. Specifically, we define the input space as $\mathcal{X}^p = \mathcal{X} \times \ldots \mathcal{X}$, where $\mathcal{X} \subset L^2(\Omega_x)$ is a function space over a compact and convex domain $\Omega_x \subseteq \mathbb{R}^d$.

The output space is assumed as $\mathcal{Y} = L^2(\Omega_y)$, the Hilbert space of square-integrable functions. Without loss of generality, we assume $\Omega_y = [0, 1]$. We denote the inner product and norm in L^2 by $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$, respectively. Let $I_{\mathcal{Y}}$ denote the identity operator on $\mathcal{Y}$, $\mathcal{L}(\mathcal{A}, \mathcal{B})$ denote the space of bounded linear operators from $\mathcal{A}$ to $\mathcal{B}$, and $\mathcal{L}(\mathcal{A})$ denote the space of bounded linear operators from $\mathcal{A}$ to itself. For an operator L, we use L' to denote its adjoint, and define its determinant $|L|$ as the product of its eigenvalues. We use $\mathbb{N}_+$ to denote the set of positive integers and $\mathbb{R}_+$ for the set of positive real numbers.

Function-on-Function Gaussian Process

In this section, we first introduce the proposed FFGP based on a separable operator-valued kernel function. We then present the construction and computational details of the operator-valued kernel, followed by a discussion of parameter estimation methods.

Definition

We define the prior of f as an FFGP:

$$f(\cdot) \sim \mathcal{FFGP}\left(\mu, K(\cdot, \cdot)\right), \tag{4}$$

where $\mu \in \mathcal{Y}$ is the prior mean function. The correlation function $K : \mathcal{X}^p \times \mathcal{X}^p \to \mathcal{L}(\mathcal{Y})$ is an operator-valued kernel (Kadri et al. 2016), defined as follows.

Definition 1 *A function $K : \mathcal{X}^p \times \mathcal{X}^p \to \mathcal{L}(\mathcal{Y})$ is called an operator-valued kernel if it satisfies:*
(1) Hermitian symmetry: $\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}^p$, $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = K'(\boldsymbol{x}_j, \boldsymbol{x}_i)$.
(2) Positive semi-definiteness: $\forall \boldsymbol{x}_i \in \mathcal{X}^p$, $y_i \in \mathcal{Y}$, $m \in \mathbb{N}_+$, $\sum_{i,j=1}^{m} \langle y_i, K(\boldsymbol{x}_i, \boldsymbol{x}_j) y_j \rangle \geqslant 0$.

The construction of K will be discussed in detail later. Given the initial dataset $\mathcal{D}_0 = \{\mathbf{X}_n, \boldsymbol{Y}_n\}$, the posterior distribution of f at a new input $\boldsymbol{x} \in \mathcal{X}^p$ is a functional Gaussian distribution with posterior mean:

$$\hat{f}(\boldsymbol{x}; \mathcal{D}_0) = \mu + \boldsymbol{K}_n(\boldsymbol{x})^\top \left(\mathbf{K}_n + \tau^2 I_{\mathcal{Y}}\right)^{-1} \left(\boldsymbol{Y}_n - \mathbf{1}_n \mu\right), \tag{5}$$

and covariance operator:

$$\hat{K}(\boldsymbol{x}, \boldsymbol{x}; \mathcal{D}_0) = K(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{K}_n(\boldsymbol{x})^\top \left(\mathbf{K}_n + \tau^2 I_{\mathcal{Y}}\right)^{-1} \boldsymbol{K}_n(\boldsymbol{x}), \tag{6}$$

where $\mathbf{1}_n$ is an $n \times 1$ vector of ones, $\boldsymbol{K}_n(\boldsymbol{x}) \in \mathcal{L}(\mathcal{Y})^n$ is an $n \times 1$ vector whose the i-th element $K(\boldsymbol{x}_i, \boldsymbol{x})$, and $\mathbf{K}_n = \{K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)\}_{i,j=1}^n$ is a $n \times n$ block operator matrix.

Operator-Valued Kernel

Selecting an appropriate operator-valued kernel is key to constructing the FFGP. While traditional GPs use matrix-valued kernels to capture correlations among multivariate outputs, operator-valued kernels generalize this idea to infinite-dimensional output spaces. According to Definition 1, one practical approach is to first define a scalar kernel $k_x : \mathcal{X}^p \times \mathcal{X}^p \to \mathcal{X}^\star$, and then apply an operator-valued map $T_{\mathcal{Y}}$: $\mathcal{X}^\star \to \mathcal{L}(\mathcal{Y})$ (Kadri et al. 2016). To ensure computational feasibility and invertibility of $\mathbf{K}_n$, we assume $\mathcal{X}^\star \subseteq \mathbb{R}_+$ and take k_x to be a positive definite scalar kernel. The resulting operator valued kernel then adopts a separable form:

$$K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \sigma^2 k_x(\boldsymbol{x}_i, \boldsymbol{x}_j) T_{\mathcal{Y}}, \ \ \forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}^p, \tag{7}$$

where $T_{\mathcal{Y}} \in \mathcal{L}_{\mathcal{Y}}$ is a nonnegative self-adjoint operator. This separable structure is widely used in multivariate-output GP models (Santner et al. 2003; Han and Ouyang 2021).

To construct a stationary FFGP, we define the scalar kernel k_x as a function of the L^2 norm between function-valued inputs, a widely used metric in FDA (Cuevas, Febrero, and Fraiman 2004; Zhang and Chen 2007). The choice of k_x depends on the characteristics of the data. In this work, we adopt the widely used Matérn kernel with variance parameter $\sigma^2 \in \mathbb{R}_+$, range parameter $\boldsymbol{\psi}_x \in \mathbb{R}_+^p$, and smoothness parameter $\nu \in \mathbb{R}_+$, as detailed in Appendix A in the extended version. To model output correlations, we define $T_{\mathcal{Y}}$ as a Hilbert-Schmidt integral operator with an exponential kernel:

$$\left(T_{\mathcal{Y}} y\right)(t) = \int_{\Omega_y} k_y(s, t) y(s) ds, \tag{8}$$

where k_y is a positive function with a range parameter $\psi_y \in \mathbb{R}_+$. Common choices include Exponential kernel and Wiener kernel functions. It can be verified that $T_{\mathcal{Y}}$ is self-adjoint and positive. Other choices of k_x and $T_{\mathcal{Y}}$ can be adopted depending on application-specific properties (see Appendix A in the extended version).

A key challenge lies in inverting the block operator matrix $\mathbf{K}_n$. We address this by employing the eigen-decomposition of $T_{\mathcal{Y}}$, as stated in the following lemma (Naylor and Sell 1982):

Lemma 1 *Let $T_{\mathcal{Y}} \in \mathcal{L}(\mathcal{Y})$ be a compact and normal operator. Then there exists an orthonormal basis $\{v_i, i \in \mathbb{N}_+\}$ and corresponding eigenvalues $\{\beta_i, i \in \mathbb{N}_+\}$, such that for any $y \in \mathcal{Y}$, $T_{\mathcal{Y}} y = \sum_{i=1}^{\infty} \beta_i \langle y, v_i \rangle v_i$.*

This spectral structure yields the decomposition $K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) y = \sum_{l=1}^{\infty} k_x\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \beta_l \langle y, v_l \rangle v_l$. When k_y is the exponential kernel or the Wiener kernel functions, the closed forms eigen-decomposition of T_y are provided in Appendix B in the extended version.

The block operator kernel matrix is represented as a Kronecker product $\mathbf{K}_x^{(n)} \otimes T_{\mathcal{Y}}$, where $\mathbf{K}_x^{(n)}$ is the $n \times n$ matrix with the ij-th entry $k_x(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Furthermore, we conduct eigen-decompsotion on $\mathbf{K}_x^{(n)}$, and denote $\{\boldsymbol{w}_j, \alpha_j\}_{j=1}^n$ as the corresponding eigenvectors and eigenvalues. Then, the inverse operator has the expansion $(\mathbf{K}_n + \lambda I_{\mathcal{Y}})^{-1} \boldsymbol{Y}_n = \sum_{i=1}^{\infty} \sum_{j=1}^{n} 1/(\alpha_j \beta_i + \lambda) \langle \boldsymbol{Y}_n, \boldsymbol{w}_j v_i \rangle \boldsymbol{w}_j v_i$, where $\langle \boldsymbol{Y}_n, \boldsymbol{w}_j v_i \rangle = \sum_{l=1}^{n} \langle y_l, w_{jl} v_i \rangle$ and w_{jl} is the l-th element of $\boldsymbol{w}_j$. Thus, the posterior mean at a new input $\boldsymbol{x} \in \mathcal{X}^p$ becomes:

$$\hat{f}(\boldsymbol{x}; \mathcal{D}_0) = \mu + \boldsymbol{k}_x^{(n)}(\boldsymbol{x})^\top \sum_{i=1}^{\infty} \sum_{j=1}^{n} \eta_{ij} \boldsymbol{w}_j v_i, \tag{9}$$

with the covariance operator:

$$\left(\hat{K}(\boldsymbol{x},\boldsymbol{x};\mathcal{D}_0)y\right)(t) = \sigma^2 \sum_{i=1}^{\infty} \delta_i \langle y, v_i\rangle v_i(t), \forall y \in \mathcal{Y}, \tag{10}$$

where $\boldsymbol{k}_x^{(n)}(\boldsymbol{x}) = k_x(\mathbf{X}_n, \boldsymbol{x})$ is a $n \times 1$ vector, $\eta_{ij} = \frac{\beta_i}{\alpha_j\beta_i+\tau^2}\langle \boldsymbol{Y}_n - \mathbf{1}_n\mu, \boldsymbol{w}_j v_i\rangle$, and $\delta_i = \beta_i\big[k_x(\boldsymbol{x},\boldsymbol{x}) - \sum_{j=1}^{n} \boldsymbol{k}_x^{(n)}(\boldsymbol{x})^\top \boldsymbol{w}_j \boldsymbol{w}_j^\top \boldsymbol{k}_x^{(n)}(\boldsymbol{x})\big]$ is the i-th eigenvalue of $\hat{K}(\boldsymbol{x},\boldsymbol{x};\mathcal{D}_0)$. The detailed computations are provided in Appendix C in the extended version.

It is noted that to balance computational efficiency and predictive accuracy, we approximate the infinite expansion by truncating the first $m \in \mathbb{N}_+$ eigenpairs. For example, the truncation level m can be chosen such that at least 90% of the total trace of the prior covariance matrix is captured,

$$m = \underset{m\in\mathbb{N}_+}{\arg\min} \left\{ \sum_{l=1}^{m} \beta_l / \sum_{l=1}^{\infty} \beta_l \geq 90\% \right\}. \tag{11}$$

Consequently, the approximated forms of $\hat{f}(\boldsymbol{x};\mathcal{D}_0)$ and $\hat{K}(\boldsymbol{x},\boldsymbol{x};\mathcal{D}_0)$ are:

$$\hat{f}_m(\boldsymbol{x};\mathcal{D}_0) = \mu + \boldsymbol{k}_x^{(n)}(\boldsymbol{x})^\top \sum_{i=1}^{m}\sum_{j=1}^{n} \eta_{ij}\boldsymbol{w}_j v_i, \tag{12}$$

$$\left(\hat{K}_m(\boldsymbol{x},\boldsymbol{x};\mathcal{D}_0)y\right)(t) = \sigma^2 \sum_{i=1}^{m} \delta_i \langle y, v_i\rangle v_i(t), \tag{13}$$

Remark 1 *The truncation level m can be adjusted depending on the specific problem context. For high-fidelity models that require higher accuracy, a larger m may be preferred. For low-fidelity models that emphasize computational efficiency, a smaller m may suffice. In practice, m can also be treated as a tuning parameter and selected via cross-validation.*

Parameter Estimation

In practice, based on the initial dataset $\mathcal{D}_0$, we directly estimate μ as the empirical mean of these initial outputs. We define the set of other parameters to be estimated as $\boldsymbol{\Theta} = \{\sigma^2, \tau^2, \boldsymbol{\psi}_x, \psi_y, \nu\}$. As noted by Zhang (2004), jointly estimating $\boldsymbol{\psi}_x$ and ν can lead to mutual non-identifiability under infill asymptotics. To address this issue, Zhang (2004) and Kaufman and Shaby (2013) recommend fixing ν and estimating the remaining parameters. Following this guideline, we fix $\nu = 5/2$ and estimate $\{\sigma^2, \tau^2, \boldsymbol{\psi}_x, \psi_y\}$ via integrated maximum likelihood. Let $\boldsymbol{\Sigma}_{\boldsymbol{Y}}^{(n)} = \mathbf{K}_n + \tau^2 I_{\mathcal{Y}}$. The parameters are estimated by minimizing the negative integrated log-likelihood

$$l(\boldsymbol{\Theta}) = \log\left|\boldsymbol{\Sigma}_{\boldsymbol{Y}}^{(n)}\right| + \left\langle \boldsymbol{Y}_n, \left(\boldsymbol{\Sigma}_{\boldsymbol{Y}}^{(n)}\right)^{-1}\boldsymbol{Y}_n \right\rangle. \tag{14}$$

The detailed algorithm for modeling FFGP is summarized in Algorithm 1.

Algorithm 1: Parameter estimation for training FFGP

Input: Observed data $\mathbf{X}_n$ and $\boldsymbol{Y}_n$, initialized $\boldsymbol{\Theta}_0 = \{\sigma_0^2, \tau_0^2, \boldsymbol{\psi}_{x0}, \psi_{y0}, \nu = 5/2\}$.

Initialize: $\sigma^2 \leftarrow \sigma_0^2$, $\tau^2 \leftarrow \tau_0^2$, $\boldsymbol{\psi}_x \leftarrow \boldsymbol{\psi}_{x0}$, $\psi_y \leftarrow \psi_{y0}$.

1: **while** σ^2, τ^2, $\boldsymbol{\psi}_x$, ψ_y have not converged **do**
2: Eigen-decomposition $\mathbf{K}_x^{(n)}$. Let $\alpha_j \in \mathbb{R}$ be the j-th eigenvalue, and let $\boldsymbol{w}_j \in \mathbb{R}^n$ be the j-th eigenvector, $j = 1, \ldots, n$.
3: Eigen-decomposition $T_{\mathcal{Y}}$. Let $\beta_i \in \mathbb{R}$ be the i-th eigenvalue, and $v_i \in \mathcal{Y}$ be the i-th eigenfunction. Select m by (11).
4: Update $\sigma^2, \tau^2, \boldsymbol{\psi}_x, \psi_y$ by minimizing (14) by minimizing (14) through the L-BFGS method.
5: **end while**

Remark 2 *Let N_{mc} be the computational cost of approximating the L^2 norm, and let $\mathcal{O}(n^\omega)$ denote the cost of computing the eigenpairs of $\mathbf{K}_x^{(n)}$, where $2 < \omega < 2.376$. Assume that the eigen-decomposition of $T_{\mathcal{Y}}$ has unit cost. Then the cost of computing the gradient of the log-likelihood with respect to $\boldsymbol{\Theta}$ is $\mathcal{O}(N_{mc}n^2m + (p+3)mn)$. Since the L-BFGS optimizer typically requires $\mathcal{O}(\log n)$ iterations (Bottou 2010), the overall computational complexity of Algorithm 1 is $\mathcal{O}\left(\log n\left(N_{mc}mn^2 + n^\omega + (p+3)mn\right)\right)$.*

Bayesian Optimization

Building upon the proposed FFGP surrogate for f, we develop a UCB-based optimization algorithm to sequentially select the next T function-valued inputs $\{\boldsymbol{x}_{n+1}, \ldots, \boldsymbol{x}_{n+T}\}$ based on the initial design $\mathcal{D}_0$. To this end, we first construct a UCB acquisition function for the scalarized objective $L_\phi f$ defined in (1). We then introduce a functional gradient ascent algorithm to efficiently solve the resulting optimization problem in the function space.

Acquisition Function

Denote $g_\phi(\boldsymbol{x}) := L_\phi f(\boldsymbol{x})$ as the scalarized objective in (2), where L_ϕ is a linear operator. Due to the linearity of GPs, the induced prior of $g_\phi(\boldsymbol{x})$ is a scalar-valued GP with mean $\mu^g = \int_{\Omega_y}\phi(t)\mu(t)dt$ and covariance function $k^g(\boldsymbol{x},\boldsymbol{x}') = ck_x(\boldsymbol{x},\boldsymbol{x}')$, where $c = \int_{\Omega_y}\int_{\Omega_y}\phi(t)\phi(s)k(t,s)dtds$ is a constant.

Given the initial design $\mathcal{D}_0$, the posterior distribution of g_ϕ at a new input $\boldsymbol{x} \in \mathcal{X}^p$ is Gaussian with mean

$$\hat{\mu}_g(\boldsymbol{x};\mathcal{D}_0) = L_\phi\mu + \boldsymbol{k}_x^{(n)}(\boldsymbol{x})\left(\mathbf{K}_x^{(n)} + \frac{\tau^2}{c}\mathbf{I}\right)^{-1} L_\phi(\boldsymbol{Y}_n - \mathbf{1}_n\mu^g), \tag{15}$$

and covariance

$$\hat{k}_g(\boldsymbol{x},\boldsymbol{x};\mathcal{D}_0) = c\left[k_x(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{k}_x^{(n)}(\boldsymbol{x})^\top\left(\mathbf{K}_x^{(n)} + \frac{\tau^2}{c}\mathbf{I}\right)^{-1}\boldsymbol{k}_x^{(n)}(\boldsymbol{x})\right]. \tag{16}$$

Note that since L_ϕ is a linear bounded operator, the posterior distribution of g_ϕ is equivalent to applying L_ϕ to the posterior distribution of f. It establishes a theoretical equivalence between the FFGP model and the scalar-valued GP defined over g_ϕ.

We now define the UCB acquisition function based on the posterior distribution of $g_\phi(\boldsymbol{x})$. At round $t+1$, given the current dataset $\mathcal{D}_t = \{\mathbf{X}_{n+t}, \boldsymbol{Y}_{n+t}\}$, the UCB criterion is defined as:

$$\alpha_{UCB}(\boldsymbol{x} \mid \mathcal{D}_t) = \hat{\mu}_g(\boldsymbol{x}; \mathcal{D}_t) + u_t^{1/2} \sqrt{\hat{k}_g(\boldsymbol{x}, \boldsymbol{x}; \mathcal{D}_t)}, \quad (17)$$

where u_t is the tuning parameter that controls the exploration-exploitation trade-off. The next function-valued input is then selected as:

$$\boldsymbol{x}_{n+t+1} = \arg\max_{\boldsymbol{x} \in \mathcal{X}^p} \alpha_{UCB}(\boldsymbol{x} \mid \mathcal{D}_t). \quad (18)$$

Functional Gradient Ascent Algorithm

Since α_{UCB} is generally nonlinear in $\boldsymbol{x}$, its maximization cannot be solved analytically in closed form. We therefore develop a functional gradient ascent (FGA) algorithm by using the Fréchet derivative, which generalizes classical derivatives to Banach spaces (Chae 2020).

Definition 2 *Let U and V be Banach spaces, and let $W \subset U$ be an open set. A map $F : W \to V$ is said to be Fréchet differentiable at a point x in W if there exists a bounded linear operator $A \in \mathcal{L}(U, V)$ such that for any $x \in U$ and $x + h \in W$, we have $F(x+h) = F(x) + A(h) + o(\|h\|)$ as $h \to 0$. The operator A is the Fréchet derivative of F at x, denoted as $\nabla F \mid_x$.*

In this work, the input $\boldsymbol{x} = (x_1, \ldots, x_p)$ is a vector of functions. The Fréchet derivative can be naturally extended by computing derivatives with respect to each component x_i, $i = 1, \ldots, p$. According to the chain rule, $\nabla \alpha_{UCB} \mid_{\boldsymbol{x}}$ only requires computing the Fréchet derivative of $k_x(\boldsymbol{x}_i, \boldsymbol{x})$, denoted as $\nabla k_x(\boldsymbol{x}_i, \boldsymbol{x}) \mid_{\boldsymbol{x}}$ for $i = 1, \ldots, n+t-1$. Then,

$$\nabla \alpha_{UCB} \mid_{\boldsymbol{x}} (\boldsymbol{x} \mid \mathcal{D}_t) = g_1(\boldsymbol{x}) + g_2(\boldsymbol{x}), \quad (19)$$

where

$$g_1(\boldsymbol{x}) = \nabla \boldsymbol{k}_x^{(n+t)} \mid_{\boldsymbol{x}}^\top \left(\mathbf{K}_x^{(n+t)} + \frac{\tau^2}{c} \mathbf{I} \right)^{-1} L_\phi(\boldsymbol{Y}_n - \mathbf{1}_n \mu^g),$$

$$g_2(\boldsymbol{x}) = -2c \nabla \boldsymbol{k}_x^{(n+t)} \mid_{\boldsymbol{x}}^\top \left(\mathbf{K}_x^{(n+t)} + \frac{\tau^2}{c} \mathbf{I} \right)^{-1} \boldsymbol{k}_x^{(n+t)}(\boldsymbol{x}).$$

Here, $\nabla \boldsymbol{k}_x^{(n+t)} \mid_{\boldsymbol{x}}$ is a $(n+t) \times p$ function-valued matrix whose the i-th row is $\nabla k_x(\boldsymbol{x}_i, \boldsymbol{x}) \mid_{\boldsymbol{x}}$. Detailed computations of the functional gradient are provided in Appendix D in the extended version.

To solve the maximization problem in (18), we apply the FGA update rule iteratively. Starting from an arbitrarily initial function $\boldsymbol{x}^{\{0\}}$, we update

$$\boldsymbol{x}_t^{\{l\}} = \boldsymbol{x}_t^{\{l-1\}} - \gamma \nabla \alpha_{UCB} \mid_{\boldsymbol{x}} \left(\boldsymbol{x}_t^{\{l-1\}} \mid \mathcal{D}_t \right), \quad (20)$$

where $\gamma \in \mathbb{R}_+$ is the step size and $l \geq 1$. The detailed algorithm for the proposed FFBO using FGA procedure using FGA is summarized below.

Algorithm 2: FFBO based on FGA

Input: Sequential data size T, initial data set $\mathcal{D}_0 = \{\mathbf{X}_n, \boldsymbol{Y}_n\}$, initial parameters $\boldsymbol{\Theta}_0$;

1: **for** round $t = 0, \ldots, T$ **do**
2: Update the posterior mean in (15) and covariance in (16) of g_ϕ based on $\boldsymbol{\Theta}_{n+t}$;
3: Obtain the next function-valued input $\boldsymbol{x}_{n+t+1}$ based on (18)-(20);
4: Run the function f and obtain the corresponding observed output $y_{n+t+1} \in \mathcal{Y}$;
5: Update $\mathcal{D}_{n+t+1} \leftarrow \mathcal{D}_{n+t} \cup \{\boldsymbol{x}_{n+t+1}, y_{n+t+1}\}$;
6: Update $\boldsymbol{\Theta}_{t+1} = \{\tau_{t+1}^2, \sigma_{t+1}^2, (\boldsymbol{\psi}_x)_{t+1}, (\psi_y)_{t+1}\}$ by minimizing (14) based on Algorithm 1;
7: **end for**
8: Obtain $i^\star = \arg\max_{i \in [n+T+1]} g_\phi(\boldsymbol{x}_i)$;
9: **Output:** The optimal point $\boldsymbol{x}_{i^\star}$ and the corresponding output $y_{i^\star}$.

Remark 3 *When using the FGA algorithm to select the optimal function-valued input $\boldsymbol{x}_{n+t+1}$ at round $t+1$, for any initial function, the computational complexity of the derivative of $\alpha_{UCB}(\boldsymbol{x} \mid \mathcal{D}_t)$ with respect to $\boldsymbol{x}$ is $\mathcal{O}\left(N_{mc} m^2 (n+t)^2 + p(n+t)^2\right)$, and the computational complexity of updating $\alpha_{UCB}(\boldsymbol{x})$ is $\mathcal{O}\left(N_{mc}^2 m^2 (n+t)^2\right)$. Therefore, the computational complexity of Algorithm 2 is $\mathcal{O}\left(N_{mc}^2 m^2 (n+t)^2 + p(n+t)^2\right)$.*

Theoretical Properties

This section begins with fundamental definitions and assumptions, upon which we build the theoretical analysis of the proposed framework. We first establish properties of the FFGP model under the following assumptions.

Assumption 1 *The underlying function f is assumed to be an FFGP with mean function $\mu \in \mathcal{Y}$ and covariance operator $K(\cdot, \cdot) = k_x(\cdot, \cdot) \otimes T_{\mathcal{Y}} \in \mathcal{L}(\mathcal{Y})$ which has a separable structure. k_x is a Matérn kernel with smoothness parameter $\nu = 5/2$, and $T_{\mathcal{Y}}$ is an integral operator with k_y corresponding to an exponential covariance function.*

Assumption 2 *The operator $T_{\mathcal{Y}} \in \mathcal{L}(\mathcal{Y})$ is a trace class, i.e. $\exists C_0 > 0$, s.t. $\sum_{i=1}^{\infty} \beta_i < C_0$.*

Assumption 3 *$\forall y \in \mathcal{Y}$, $\exists C_1 > 0$, s.t. $\|y\| \leq C_1$.*

Assumption 4 *$\exists 0 < c < 1$ s.t. $\frac{\tau^2}{\sigma^2} \asymp n^{-c}$, i.e., $\exists 0 < a < b$ s.t. $a < \left(\frac{\tau^2}{\sigma^2}\right) / n^{-c} < b$ as $n \to \infty$, where τ^2 is the noise variance and σ^2 is the process variance.*

Assumption 1 ensures that the true function f is a well-specified FFGP, which is a basic setting in the GP regression framework. Assumption 2 guarantees that functional random variables in the FFGP model are measurable in $\mathcal{Y}$. The boundedness condition in Assumption 3 $\mathcal{Y} = L^2[0,1]$. The convergence rate of τ^2/σ^2 in Assumption 4 ensures the convergence of the approximation error between $\hat{f}_m$ and $\hat{f}$. Under these assumptions, we establish the following properties of the FFGP model.

Theorem 1 *Under Assumptions 1-4, the operator-valued conditional covariance* $\mathrm{Var}\left[f(\boldsymbol{x}) \mid \boldsymbol{Y}\right]$ *in (6) is nonnegative and a trace class. Then* $\hat{f}(\boldsymbol{x})$ *defines a measure on* $\mathcal{Y}$ *(i.e. a* $\mathcal{Y}$*-valued random variable).*

Its proof details are shown in Appendix E in the extended version. Theorem 1 shows for any input over $\mathcal{X}^p$, the posterior density of FFGP is measurable.

Theorem 2 *Under Assumptions 1-4, for any input* $\boldsymbol{x} \in \mathcal{X}^p$, $\|\hat{f}_m(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})\| \leq C_5 m^{-1} \to 0$ *as* $m \to \infty$.

Its proof details are shown in Appendix F in the extended version. Theorem 2 shows that the error between the truncated predictor and the predictor of FFGP goes to zero when $m \to \infty$, and its convergence rate is $O\left(m^{-1}\right)$.

Next, we analyze the theoretical properties of the proposed FFBO algorithm. To be general, we set the initial design $\mathcal{D}_0 = \varnothing$, that is, $n = 0$. We first define the notion of regret for the FFBO algorithm, which measures the performance loss caused by not knowing the objective function $g_\phi(\boldsymbol{x})$ in advance. A desirable learning strategy aims to achieve sublinear cumulative regret, meaning that the average regret per round tends to zero as the number of rounds increases.

Definition 3 *At each round* t*, the FFBO method selects a queried function-valued input* $\boldsymbol{x}_t \in \mathcal{X}^p$*. The simple regret is defined as* $r_t = [g_\phi(\boldsymbol{x}^\star) - g_\phi(\boldsymbol{x}_t)]$*, and the cumulative regret up to round* T *is* $R_T = \sum_{t=1}^{T} [g_\phi(\boldsymbol{x}^\star) - g_\phi(\boldsymbol{x}_t)]$*, where* $\boldsymbol{x}^\star$ *is the optimizer defined in (2).*

Then we introduce the following assumptions and lemmas related to the regret bound of the FFBO framework.

Assumption 5 *For every* $\boldsymbol{x} = (x_1, \cdots, x_p)^\top \in \mathcal{X}^p$*, the components satisfy* $\|x_j\|_{\mathcal{N}_\Phi(\Omega_x)} \leq 1$*, where* $\mathcal{X} = \mathcal{N}_\Phi(\Omega_x)$ *is a reproducing kernel Hilbert space (RKHS) generated by the kernel* Φ *with norm* $\|\cdot\|_{\mathcal{N}_\Phi(\Omega_x)}$*,* $j = 1, \cdots, p$.

Assumption 6 *The Fréchet derivative of* g_ϕ *with respect to* $\boldsymbol{x} \in \mathcal{X}^p$ *is bounded with high probability. That is, there exist constants* $a, b \in \mathbb{R}_+$*, such that*

$$\Pr\left\{\sup_{\boldsymbol{x}\in\mathcal{X}^p} \left\|\frac{\partial f}{\partial x_j}\right\| > L\right\} \leq a e^{-(L/b)^2},\ j = 1, \ldots, p. \tag{21}$$

Assumption 5 ensures that the input $\mathbf{x}$ is bounded in an RKHS. Assumption 6 states that the gradient with respect to each input component is uniformly bounded over the input space with an exponentially decaying tail.

Lemma 2 *Under Assumptions 1-6, for any* $\delta \in (0, 1)$*, and* $u_t = \log(2p\pi_t/\delta) + 2p \log\left(pbt^2\pi_t\sqrt{\log(2pa/\delta)}\right)$*, where* $\sum_{t=1}^{\infty} 1/\pi_t = 1$ *with* $\pi_t \in \mathbb{R}_+$*. Then, for all* $\boldsymbol{x} \in \mathcal{X}^p$*,*

$$|g_\phi(\boldsymbol{x}) - \hat{\mu}_g(\boldsymbol{x}; \mathcal{D}_{t-1}| \leq u_t^{1/2}\sqrt{\hat{k}_g(\boldsymbol{x}, \boldsymbol{x}; \mathcal{D}_{t-1})} \tag{22}$$

holds with high probability.

Its proof is given in Appendix G in the extended version. Lemma 2 provides a concentration inequality for the objective g_ϕ, showing that the GP posterior concentrates tightly around its posterior mean with high probability.

Theorem 3 *Under Assumptions 1-6, for any* $\delta \in (0, 1)$*, the FFBO method ensures that, with probability at least* $1 - \delta$*, the cumulative regret over* N *rounds is bounded by*

$$\Pr\left\{R_T \leq \sqrt{B_1 T u_T \gamma_T} + \pi^2/6 \quad \forall T \geq 1\right\} \geq 1 - \delta, \tag{23}$$

where $B_1 = 8/\log(1 + \sigma^2/\tau^2)$*, and* γ_T *is the information gain after* T *rounds.*

Its proof details are shown in Appendix H in the extended version. Theorem 3 shows that the regret of the proposed FFBO is sub-linear and achieves $\mathcal{O}^\star(\sqrt{T})$ regret.

Experiments

In this section, we evaluate the effectiveness and performance of our proposed FFBO framework through three synthetic experiments and a real-world case study. We compare FFBO against the following three baseline methods from the literature: (1) function-valued input Bayesian optimization (FIBO) (Vien, Zimmermann, and Toussaint 2018), which considers function valued inputs in a RKHS and scalar valued outputs, and uses the integral of the outputs as the optimization objective; (2) function-valued output Bayesian optimization (FOBO) (Huang et al. 2021), which considers vector-valued inputs and function-valued outputs, where the optimization objective is defined as the L_2-norm of the function-valued output; (3) multi-task Bayesian optimization (MTBO) (Chowdhury and Gopalan 2021), which considers vector-valued inputs and vector-valued outputs, and applies a weighted scalarization of the outputs as the optimization objective. A detailed description of all baselines is provided in Appendix I in the extended version.

Synthetic Experiments

The following three synthetic scenarios are designed to evaluate the performance of the proposed FFBO algorithm. In all settings, the objective is to maximize $g_\phi(\boldsymbol{x})$.

- **Setting 1**: The function-valued input is defined as $x(s) = a\cos(\pi/[b + \sin(\exp(s) + \pi s)^2])$, where $(a, b) \in [0.5, 1.5]^2$. The corresponding output is $f(x)(t) = -(3 + \sin(2\pi t))\int_0^1 (x(s) - x_0(s))^2 ds$, where $x_0(s) = \cos(\pi/[1 + \sin(\exp(s_0) + \pi s_0)^2])$. The objective is defined as $g_\phi(x) = \int_{\Omega_y} f(x)(t)dt$. The optimal input is $x^\star(s) = x_0(s)$, yielding the optimal value $g_\phi(x^\star) = 0$.
- **Setting 2**: Following Setting 1, the function-valued input $x(s)$ and $x_0(s)$ are the same as in Setting 1. The output is defined as $f(x) = -3\exp(\int_0^1 (x(s) - x_0(s))^2 ds)$, and the objective is $g_\phi(x) = \int_{\Omega_y} f(x)(t)dt$. The optimum is achieved at $x^\star(s) = x_0(s)$, with the corresponding maximum value $g_\phi(x^\star) = -3$.

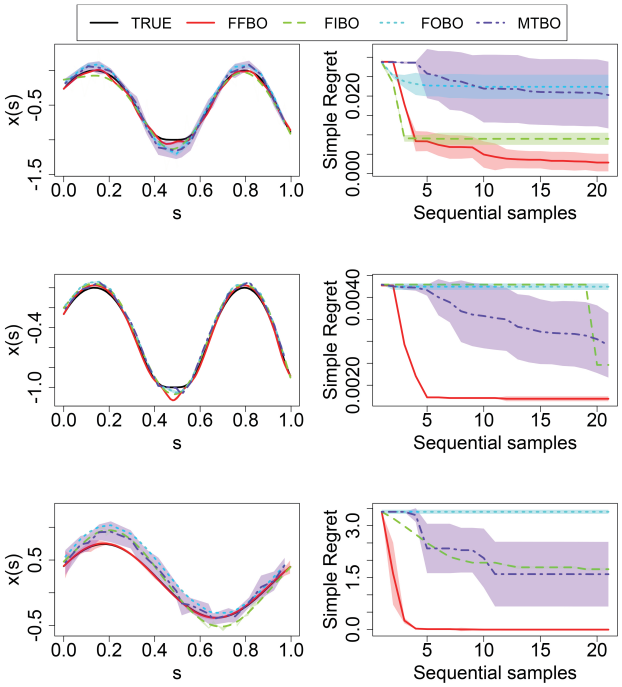

Figure 1: The optimal input (Left) and each round's simple regret (Right) of different methods in Settings 1 (Top panel), Setting 2 (Median panel), and Setting 3 (Bottom panel).

- **Setting 3**: The function-valued input is constructed as a linear combination of Fourier basis functions $x(s) = \theta_1 \sin(2\pi s) + \theta_2 \cos(2\pi s) + \theta_3 \exp(-5(s-0.5)^2)$, where $(\theta_1, \theta_2, \theta_3) \in [0.01, 0.99]^3$. The function-valued output is given by $f(x,t) = 20\exp(-5\int_0^1 (x(s) - x_0(s))^2 ds) + 10\sin(3\pi t)\int_0^1 x(s)\sin(3\pi s)ds$, where the optimal input is $x_0 = \frac{1}{2}\sin(2\pi s) + \frac{1}{3}\cos(2\pi s) + \frac{1}{4}\exp(-5(s-0.5)^2)$. The objective function is defined as $g_\phi(x) = \int_{\Omega_y} f(x)(t)dt$. The maximum is obtained at $x^\star(s) = x_0(s)$, with $g_\phi(x^\star) = 20.2899$.

We use the Matérn kernel with smoothness parameter $5/2$ to model input correlations for all baseline methods, and the exponential kernel to model output correlations. For the proposed FFBO method, we set the learning rate as $\gamma = 0.01/l$, where l denotes the current iteration number. An initial design of $n = 10$ input-output pairs is randomly generated, with observation noise variance $\tau^2 = 0.01^2$. Then, each method sequentially selects $T = 20$ additional input samples. The final objective value $g_\phi(\boldsymbol{x}^\star)$ obtained after these evaluations is reported as the best solution. For fairness, all methods are evaluated under ten independent replicates.

Figure 1 shows the results in terms of the optimal input trajectory and the evolution of the simple regret over 20 iterations for each of the three settings. As shown in Figure 1, the proposed FFBO method consistently outperforms all baseline methods across all settings. FIBO cannot directly handle function-valued outputs and thus relies on discretization,

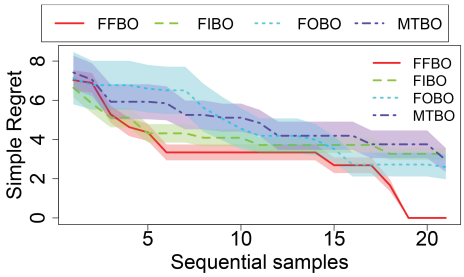

Figure 2: Each round's simple regret of different methods.

which reduces estimation accuracy and limits its ability to identify the optimum. Similarly, FOBO and MTBO are not designed to handle function-valued inputs, leading to poor performance and non-smooth optimal inputs.

In Appendix J in the extended version, we provide additional results for a setting involving multi-dimensional function-valued inputs, as well as extended experiments for all settings under different noise levels.

Case Study

We apply the proposed FFBO framework to a real-world dataset involving three-dimensional printed aortic valves made from tissue-mimicking metamaterials (Wang et al. 2016; Chen et al. 2021). In this case, sinusoidal waveforms serve as function-valued inputs, and the corresponding stress-strain response curves act as function-valued outputs. The dataset contains 76 observations. We randomly select $n = 10$ for the initial design and choose $T = 20$ additional queries from the remaining 66 candidates. Each baseline is repeated across 10 independent runs. The simple regret across rounds is shown in Figure 2. It can be seen that the proposed FFBO framework efficiently explores the function valued input space, identifies effective waveforms, and remains robust to perturbations.

Conclusion

We proposed a novel FFBO framework to address optimization problems in which both the inputs and outputs are functions. Our method is built upon an FFGP model that operates directly in the function space using a separable operator-valued kernel, which allows us to capture complex dependencies between functional variables. Based on the FFGP, we introduced a scalar UCB acquisition function through an operator-based scalarization strategy and developed an FGA algorithm for efficient optimization in infinite-dimensional input spaces. We further analyzed the theoretical properties of the FFBO framework and validated its effectiveness on both synthetic and real-world datasets. The results show that FFBO significantly outperforms existing methods, demonstrating its potential for advancing Bayesian optimization in function-valued domains.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant No.72271138) and Tsinghua-National University of Singapore Joint Funding (Grant No.20243080039). We gratefully acknowledge this support.

References

Astudillo, R.; and Frazier, P. 2021. Bayesian optimization of function networks. *Advances in neural information processing systems*, 34: 14463–14475.

Bottou, L. 2010. Large-Scale Machine Learning with Stochastic Gradient Descent. *Proc. of COMPSTAT*.

Chae, S. B. 2020. *Holomorphy and Calculus in Normed SPates*. CRC Press.

Chen, J.; Mak, S.; Joseph, V. R.; and Zhang, C. 2021. Function-on-function kriging, with applications to three-dimensional printing of aortic tissues. *Technometrics*, 63(3): 384–395.

Chowdhury, S. R.; and Gopalan, A. 2021. No-regret algorithms for multi-task bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, 1873–1881. PMLR.

Cuevas, A.; Febrero, M.; and Fraiman, R. 2004. An anova test for functional data. *Computational statistics & data analysis*, 47(1): 111–122.

Daxberger, E.; Makarova, A.; Turchetta, M.; and Krause, A. 2020. Mixed-Variable Bayesian Optimization. In *Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 20)*, 2616–2622. Curran Associates, Inc.

Frazier, P. I. 2018. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Gultchin, L.; Aglietti, V.; Bellot, A.; and Chiappa, S. 2023. Functional causal Bayesian optimization. In *Uncertainty in Artificial Intelligence*, 756–765. PMLR.

Han, M.; and Ouyang, L. 2021. Robust functional response-based metamodel optimization considering both location and dispersion effects for aeronautical airfoil designs. *Structural and Multidisciplinary Optimization*, 64(3): 1545–1565.

Huang, C.; Ren, Y.; McGuinness, E. K.; Losego, M. D.; Lively, R. P.; and Joseph, V. R. 2021. Bayesian optimization of functional output in inverse problems. *Optimization and Engineering*, 22: 2553–2574.

Jank, W.; and Zhang, S. 2011. An automated and data-driven bidding strategy for online auctions. *INFORMS Journal on computing*, 23(2): 238–253.

Kadri, H.; Duflos, E.; Preux, P.; Canu, S.; Rakotomamonjy, A.; and Audiffren, J. 2016. Operator-valued Kernels for Learning from Functional Response Data. *Journal of Machine Learning Research*, 17(20): 1–54.

Kaufman, C.; and Shaby, B. A. 2013. The role of the range parameter for estimation and prediction in geostatistics. *Biometrika*, 100(2): 473–484.

Khan, N.; Goldberg, D. E.; and Pelikan, M. 2002. Multi-objective Bayesian optimization algorithm. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, 684–684.

Lan, T.; Li, Z.; Li, Z.; Bai, L.; Li, M.; Tsung, F.; Ketter, W.; Zhao, R.; and Zhang, C. 2023. MM-DAG: Multi-task DAG Learning for Multi-modal Data–with Application for Traffic Congestion Analysis. *ACM SIGKDD*.

Laumanns, M.; and Ocenasek, J. 2002. Bayesian optimization algorithms for multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, 298–307. Springer.

Morris, M. D. 2012. Gaussian surrogates for computer models with time-varying inputs and outputs. *Technometrics*, 54(1): 42–50.

Naylor, A. W.; and Sell, G. R. 1982. *Linear operator theory in engineering and science*. Springer Science & Business Media.

Nguyen, N. C.; and Peraire, J. 2015. Gaussian functional regression for linear partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 287: 69–89.

Paria, B.; Kandasamy, K.; and Póczos, B. 2020. A flexible framework for multi-objective bayesian optimization using random scalarizations. In *Uncertainty in Artificial Intelligence*, 766–776. PMLR.

Santner, T. J.; Williams, B. J.; Notz, W. I.; and Williams, B. J. 2003. *The design and analysis of computer experiments*, volume 1. Springer.

Shields, B. J.; Stevens, J.; Li, J.; Parasram, M.; Damani, F.; Alvarado, J. I. M.; Janey, J. M.; Adams, R. P.; and Doyle, A. G. 2021. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844): 89–96.

Snoek, J.; Larochelle, H.; and Adams, R. P. 2012. Practical Bayesian Optimization of Machine Learning Algorithms. In Pereira, F.; Burges, C.; Bottou, L.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Sung, C.-L.; Wang, W.; Cakoni, F.; Harris, I.; and Hung, Y. 2024. Functional-Input Gaussian Processes with Applications to Inverse Scattering Problems. *Statistica Sinica*.

Tuo, R.; He, S.; Pourhabib, A.; Ding, Y.; and Huang, J. Z. 2023. A reproducing kernel hilbert space approach to functional calibration of computer models. *Journal of the American Statistical Association*, 118(542): 883–897.

Vien, N. A.; Zimmermann, H.; and Toussaint, M. 2018. Bayesian functional optimization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, 4171–4178.

Wang, K.; Zhao, Y.; Chang, Y.-H.; Qian, Z.; Zhang, C.; Wang, B.; Vannan, M. A.; and Wang, M.-J. 2016. Controlling the mechanical behavior of dual-material 3D printed meta-materials for patient-specific tissue-mimicking phantoms. *Materials & Design*, 90: 704–712.

Wang, S.; Ng, S. H.; and Haskell, W. B. 2022. A multilevel simulation optimization approach for quantile functions. *INFORMS Journal on Computing*, 34(1): 569–585.

Wang, X.; Jin, Y.; Schmitt, S.; and Olhofer, M. 2023. Recent advances in Bayesian optimization. *ACM Computing Surveys*, 55(13s): 1–36.

Wang, Y.; Wang, M.; AlBahar, A.; and Yue, X. 2022. Nested Bayesian optimization for computer experiments. *IEEE/ASME Transactions on Mechatronics*, 28(1): 440–449.

Zhang, H. 2004. Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *Journal of the American Statistical Association*, 99(465): 250–261.

Zhang, J.-T.; and Chen, J. 2007. Statistical Inferences for Functional Data. *The Annals of Statistics*, 35(3): 1052–1079.